WEST Search History

DATE: Wednesday, August 13, 2003

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L18	L11	26	L18
	DB=EPAB,DWPI,TDBD; PLUR=YES; OP=ADJ		
L17	L16	23	L17
	DB=JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=ADJ		
L16	L12 and (information near (trace or debug$))	34	L16
L15	L12 and (bus near (interfac$ or link))	4	L15
L14	L12 and ((bus near communication) same processor)	0	L14
L13	L12 and (bus near (interfac$ or link) same processor)	0	L13
L12	debug$ near (circuit or chip or module)	364	L12
	DB=USPT,PGPB; PLUR=YES; OP=ADJ		
L11	L10 and exception	30	L11
L10	L9 or l8	72	L10
L9	L1 and (bus near (interfac$ or link) same processor)	66	L9
L8	L1 and ((bus near communication) same processor)	15	L8
L7	L6 and exception	13	L7
L6	L5 or l4	22	L6
L5	L2 and (bus near (interfac$ or link) same processor)	21	L5
L4	L2 and ((bus near communication) same processor)	6	L4
L3	L2 and bus near (interfac$ or link)	44	L3
L2	L1 and information near (trace or debug$)	149	L2
L1	debug$ near (circuit or chip or module)	649	L1

END OF SEARCH HISTORY

WEST

Generate Collection | Print

Search Results - Record(s) 1 through 20 of 26 returned.

☐ 1. Document ID: US 6591378 B1

L18: Entry 1 of 26 File: USPT Jul 8, 2003

US-PAT-NO: 6591378
DOCUMENT-IDENTIFIER: US 6591378 B1

TITLE: Debug controller in a data processor and method therefor

DATE-ISSUED: July 8, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Arends; John H.	Austin	TX		
Scott; Jeffrey W.	Austin	TX		
Moyer; William C.	Dripping Springs	TX		

US-CL-CURRENT: 714/38; 712/227, 714/30, 717/124

ABSTRACT:

A method for debug control in a pipelined data processor where an offset is determined for the program counter (PC) based on the state of the pipeline. The offset is subtracted from the PC value at the end of a debug session. The resultant PC value restarts fetching of a last unsuccessfully completed instruction. If the offset indicates a change to the PC value, the instruction register is adjusted to a nop to allow the pipeline to restart execution after the last successfully completed instruction. In one embodiment, the state of the machine is preserved prior to exception handling.

25 Claims, 7 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 6

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

☐ 2. Document ID: US 6557119 B1

L18: Entry 2 of 26 File: USPT Apr 29, 2003

US-PAT-NO: 6557119
DOCUMENT-IDENTIFIER: US 6557119 B1

TITLE: Microcomputer debug architecture and method

DATE-ISSUED: April 29, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Edwards; David Alan	Bristol			GB
Rich; Anthony Willis	Cambridge			NZ

US-CL-CURRENT: 714/38; 714/31, 714/47, 717/124, 717/129, 717/131

ABSTRACT:

A computer system, comprising at least one central processing unit and a memory unit coupled to the at least one central processing unit, a set of watchpoints defined in the computer system; each watchpoint in the set of watchpoints comprising a programmable precondition register and a programmable action register, a set of latches, and selection circuitry that selects one latch in the set of latches to couple an output of an action register to an input of the selected latch.

35 Claims, 23 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 23

☐ 3. Document ID: US 6542983 B1

L18: Entry 3 of 26 File: USPT Apr 1, 2003

US-PAT-NO: 6542983
DOCUMENT-IDENTIFIER: US 6542983 B1

TITLE: Microcomputer/floating point processor interface and method

DATE-ISSUED: April 1, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Gearty; Margaret Rose	Bath			GB
Peng; Chih-Jui	San Jose	CA		

US-CL-CURRENT: 712/212; 712/214, 712/215, 712/219, 712/220, 712/222

ABSTRACT:

In a computer system having a central processing unit (CPU) execution pipeline and a floating point unit (FPU) execution pipeline, the CPU execution pipeline including a CPU decoder

pipestage and the FPU execution pipeline including an FPU decoder pipestage, the method including the steps of, (a) sending a first instruction to the CPU decoder pipestage, (b) sending the first instruction to the FPU decoder pipestage, (c) generating a signal indicating that the first instruction has been accepted by the CPU decoder pipestage, (d) generating a signal indicating that the first instruction has been accepted by the FPU decoder pipestage, (e) sending a second instruction to the CPU decoder pipestage in response to step (d), and (f) sending a second instruction to the FPU decoder pipestage in response to step (c). A corresponding apparatus is also provided.

3 Claims, 7 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 6

☐ 4. Document ID: US 6502210 B1

L18: Entry 4 of 26 File: USPT Dec 31, 2002

US-PAT-NO: 6502210
DOCUMENT-IDENTIFIER: US 6502210 B1

TITLE: Microcomputer debug architecture and method

DATE-ISSUED: December 31, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Edwards; David Alan	Bristol			GB

US-CL-CURRENT: 714/38; 714/31, 714/47, 717/124, 717/129, 717/131

ABSTRACT:

A computer system including at least one central processing unit, a memory unit coupled to the at least one central processing unit, a set of watchpoints a set of watchpoints defined in the computer system, each watchpoint in the set of watchpoints including a programmable precondition register that stores a set of precondition codes, wherein the set of precondition codes is identical for each watchpoint in the set of watchpoints and a programmable action register that stores a set of action codes, wherein the set of action codes is identical for each watchpoint in the set of watchpoints, and a first comparator, having inputs coupled to the precondition register, that compares at least one precondition code in the set of precondition codes with a first data value in the computer system and provides a signal to the action register in response thereto. A method of triggering a watchpoint in a computer system is also provided.

26 Claims, 23 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 23

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 5. Document ID: US 6487683 B1

L18: Entry 5 of 26 File: USPT Nov 26, 2002

US-PAT-NO: 6487683
DOCUMENT-IDENTIFIER: US 6487683 B1

TITLE: Microcomputer debug architecture and method

DATE-ISSUED: November 26, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Edwards; David Alan	Clifton			GB

US-CL-CURRENT: 714/38; 714/31, 714/37, 714/47, 717/124, 717/129, 717/131

ABSTRACT:

A computer system, including a central processing unit and a memory unit coupled to the at least one central processing unit, a set of watchpoints defined in the computer system, each watchpoint in the set of watchpoints including a programmable precondition register that stores a set of precondition codes, wherein the set of precondition codes is identical for each watchpoint in the set of watchpoints, a programmable action register that stores a set of action codes, wherein the set of action codes is identical for each watchpoint in the set of watchpoints, a set of latches, each latch having an input and an output, and circuitry that couples at least one latch in the set of latches to at least two watchpoints in the set of watchpoints so that there is a predetermined relationship between triggering of the at least two watchpoints. A method of filtering debugging data in a computer system is also provided.

16 Claims, 23 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 23

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 6. Document ID: US 6477683 B1

L18: Entry 6 of 26 File: USPT Nov 5, 2002

US-PAT-NO: 6477683
DOCUMENT-IDENTIFIER: US 6477683 B1

TITLE: Automated processor generation system for designing a configurable processor and method for the same

DATE-ISSUED: November 5, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE ZIP CODE	COUNTRY
Killian; Earl A.	Los Altos Hills	CA	
Gonzalez; Ricardo E.	Menlo Park	CA	
Dixit; Ashish B.	Mountain View	CA	
Lam; Monica	Menlo Park	CA	
Lichtenstein; Walter D.	Belmont	MA	
Rowen; Christopher	Santa Cruz	CA	
Ruttenberg; John C.	Newton	MA	
Wilson; Robert P.	Palo Alto	CA	
Wang; Albert Ren-Rui	Fremont	CA	
Maydan; Dror Eliezer	Palo Alto	CA	

US-CL-CURRENT: 716/1; 716/18

ABSTRACT:

An automated processor design tool uses a description of customized processor instruction set extensions in a standardized language to develop a configurable definition of a target instruction set, a Hardware Description Language description of circuitry necessary to implement the instruction set, and development tools such as a compiler, assembler, debugger and simulator which can be used to develop applications for the processor and to verify it. Implementation of the processor circuitry can be optimized for various criteria such as area, power consumption, speed and the like. Once a processor configuration is developed, it can be tested and inputs to the system modified to iteratively optimize the processor implementation. By providing a constrained domain of extensions and optimizations, the process can be automated to a high degree, thereby facilitating fast and reliable development.

104 Claims, 15 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 12

☐ 7. Document ID: US 6477638 B1

L18: Entry 7 of 26 File: USPT Nov 5, 2002

US-PAT-NO: 6477638
DOCUMENT-IDENTIFIER: US 6477638 B1

TITLE: Synchronized instruction advancement through CPU and FPU pipelines

DATE-ISSUED: November 5, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Gearty; Margaret Rose	Bath			GB
Peng; Chih-Jui	San Jose	CA		

US-CL-CURRENT: 712/220; 712/203, 712/222

ABSTRACT:

A computer system having a central processing unit (CPU) execution pipeline and a floating point unit (FPU) execution pipeline, the CPU pipeline including a plurality of pipestages and the FPU pipeline including a plurality of pipestages, wherein each CPU pipestage in the CPU pipeline has a corresponding pipestage in the FPU pipeline, a method of synchronizing operation of the CPU pipeline and the FPU pipeline, the method including the steps of (a) receiving an instruction in a first CPU pipestage, (b) receiving the instruction in a corresponding first FPU pipestage, (c) processing the instruction in the first CPU pipestage, (d) processing the instruction in the first FPU pipestage, (e) generating, by the first CPU pipestage, a first signal indicating that the instruction has been processed by first CPU pipestage and is ready to proceed to a second pipestage in the CPU pipeline, (f) generating by the first FPU pipestage, a second signal indicating that the instruction has been processed by the first FPU pipestage and is ready to proceed to a second pipestage in the FPU pipeline, (g) sending the instruction from the first CPU pipestage to the second pipestage in the CPU pipeline, (h) sending the instruction from the first FPU pipestage to the second pipestage in the FPU pipeline, (i) wherein the second pipestage in the CPU pipeline responds to the second signal to send the instruction to a third pipestage in the CPU pipeline, and (j) wherein the second pipestage in the FPU pipeline responds to the first signal to send the instruction to a third pipestage in the FPU pipeline. A corresponding apparatus is also provided.

9 Claims, 7 Drawing figures
Exemplary Claim Number: 4
Number of Drawing Sheets: 6

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 8. Document ID: US 6463553 B1

L18: Entry 8 of 26 File: USPT Oct 8, 2002

US-PAT-NO: 6463553
DOCUMENT-IDENTIFIER: US 6463553 B1

TITLE: Microcomputer debug architecture and method

DATE-ISSUED: October 8, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Edwards; David Alan	Bristol			GB

US-CL-CURRENT: 714/38; 714/31, 714/47, 717/124, 717/129, 717/131

ABSTRACT:

A method of filtering debugging data in a computer system including at least one central processing unit and a memory unit coupled to the at least one central processing unit. The method includes the steps of defining a set of watchpoints in the computer system by defining a set of precondition registers and a set of action registers, defining a set of identical precondition codes to be applied to each watchpoint in the set of watchpoints, defining a set of identical action codes to be applied to each watchpoint in the set of watchpoints, storing the set of precondition codes in each precondition register in the set of precondition registers, storing the set of action codes in each action register in the set of action registers, selecting which precondition codes in the set of precondition codes are to be active for a particular watchpoint, selecting which action code in the set of action codes are to be active for a particular watchpoint, operating the computer system so as to execute a program, comparing the debugging data in the computer system with the active precondition codes for a particular watchpoint, sending a signal to the action register for the particular watchpoint when the debugging data in the computer system satisfies the active precondition codes for the particular watchpoint, and causing the computer to respond to the active action codes for the particular watchpoint.

9 Claims, 23 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 23

☐ 9. Document ID: US 6311292 B1

L18: Entry 9 of 26 File: USPT Oct 30, 2001

US-PAT-NO: 6311292
DOCUMENT-IDENTIFIER: US 6311292 B1

TITLE: Circuit, architecture and method for analyzing the operation

of a digital processing system

DATE-ISSUED: October 30, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Choquette; Jack H.	Los Altos	CA		
Smith; Donald W.	Santa Clara	CA		

US-CL-CURRENT: 714/30; 714/31

ABSTRACT:

A dual access debugging architecture. This architecture allows the microprocessor to select between external debugging, supported via the physical system interface, and internal debugging, supported via logic within the microprocessor which is controlled by decoded software instructions.

In one example of the present invention, a microprocessor includes a system bus interface and a program decoder which is coupled to the system bus interface. The system bus interface is coupled to a system bus to which external memory is coupled. Debugging operations are stored as debugging instructions in the external memory. When these debugging instructions are retrieved from memory, through the system bus and the system bus interface, they are decoded in the program decoder of the microprocessor and they in turn cause the microprocessor to enter a first debugging mode which is controlled by the debugging instructions. The first debugging mode may be referred to as an internal programmable method. The microprocessor also includes a dedicated test port, such as a JTAG port, which provides signals to and from registers and other logic in test port logic on the IC (integrated circuit) of the microprocessor. The dedicated test port includes input/output pins on the microprocessor which convey the test signals to external test logic device, such as JTAG test equipment. Testing of the microprocessor using the dedicated test port involves asserting a signal in the test port which causes the microprocessor to enter a second debugging mode which is controlled by the external test logic device. This second debugging mode may be referred to as an external debug method.

17 Claims, 6 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 6

☐ 10. Document ID: US 6289446 B1

L18: Entry 10 of 26 File: USPT Sep 11, 2001

US-PAT-NO: 6289446

DOCUMENT-IDENTIFIER: US 6289446 B1

TITLE: Exception handling utilizing call instruction with context information

DATE-ISSUED: September 11, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Nilsson; Hans-Peter	Lund			SE

US-CL-CURRENT: 712/244; 712/242

ABSTRACT:

In-code context data used for exception handling is incorporated into a special call instruction which is recognized by the processor. The information is skipped at the time of the function call and read at the time of the stack unwinding. This special call instruction may be implemented to run at no extra cycle costs compared to normal instructions, except for the external execution time dependencies from such machinery as a cache involved in the instruction fetching, since it would never be necessary during normal execution to actually access the information. The information is only accessed during exception handling.

19 Claims, 8 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 7

☐ 11. Document ID: US 6145123 A

L18: Entry 11 of 26 File: USPT Nov 7, 2000

US-PAT-NO: 6145123
DOCUMENT-IDENTIFIER: US 6145123 A

TITLE: Trace on/off with breakpoint register

DATE-ISSUED: November 7, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Torrey; James M.	Austin	TX		
Prickett; John M.	Manchaca	TX		
Lloyd; Jim L.	Austin	TX		

US-CL-CURRENT: 717/128; 703/28, 712/227, 714/30, 714/35, 714/37, 714/38, 714/45

ABSTRACT:

An information processing system such as a microprocessor includes a processor core, a debug register circuit and a trace unit. The processor core is for processing information according to a program. The program includes a plurality of instructions for execution by the processor core. Each of the plurality of instructions has a corresponding address. The debug register circuit is coupled to the processor core. The debug register circuit includes a dedicated initiate trace breakpoint register coupled to receive and store an initiate trace address and a dedicated terminate trace breakpoint register coupled to receive and store a terminate trace address. The trace unit is coupled to the debug register circuit and the processor core. The trace unit initiates a program trace responsive to the program accessing the initiate trace address. The trace unit terminates the program trace responsive to the program accessing the terminate trace address. The program trace includes information regarding the execution of the program by the microprocessor.

34 Claims, 7 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 6

☐ 12. Document ID: US 6014728 A

L18: Entry 12 of 26 File: USPT Jan 11, 2000

US-PAT-NO: 6014728
DOCUMENT-IDENTIFIER: US 6014728 A

TITLE: Organization of an integrated cache unit for flexible usage in supporting multiprocessor operations

DATE-ISSUED: January 11, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Baror; Gigy	Austin	TX		

US-CL-CURRENT: 711/133; 711/134, 711/142, 711/143, 711/144, 711/145, 711/206

ABSTRACT:

A computer system having a cache memory subsystem which allows flexible setting of caching policies on a page basis and a line basis. A cache block status field is provided for each cache block to indicate the cache block's state, such as shared or exclusive. The cache block status field controls whether the cache control unit operates in a write-through write mode or in a copy-back write

mode when a write hit access to the block occurs. The cache block status field may be updated by either a TLB write policy field contained within a translation look-aside buffer entry which corresponds to the page of the access, or by a second input independent of the TLB entry which may be provided from the system on a line basis.

7 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments KWIC | Draw Desc | Image

---

☐ 13. Document ID: US 5978937 A

L18: Entry 13 of 26 File: USPT Nov 2, 1999

US-PAT-NO: 5978937
DOCUMENT-IDENTIFIER: US 5978937 A

TITLE: Microprocessor and debug system

DATE-ISSUED: November 2, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Miyamori; Takashi	Yokohama			JP
Yano; Tatsuo	Kobe			JP

US-CL-CURRENT: 714/45; 716/4

ABSTRACT:

A microprocessor 10 has a processor core 20 and a debug module 30. The processor core 20 executes a user program and a monitor program for debugging a user target system 70. The debug module 30 serves as an interface with a debug tool 60, to let the processor core 20 execute the monitor program stored in the debug tool 60. The debug module 30 makes an interrupt or exception request to switch the processor core 20 from the user program to the monitor program.

22 Claims, 44 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 35

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments KWIC | Draw Desc | Image

---

☐ 14. Document ID: US 5680620 A

L18: Entry 14 of 26 File: USPT Oct 21, 1997

US-PAT-NO: 5680620
DOCUMENT-IDENTIFIER: US 5680620 A

TITLE: System and method for detecting access to a peripheral device using a debug register

DATE-ISSUED: October 21, 1997

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Ross; S. Timothy	Georgetown	TX		

US-CL-CURRENT: 717/129; 710/15, 714/34

ABSTRACT:

In a microprocessor, a debug facility traps accesses to a peripheral device, such as a speaker, residing at I/O port addresses. In one embodiment, a number of debug registers are provided for a system or an application program to set a trap at specific I/O or memory address, and to associate the specified address to an exception handling program. When another application program accesses the specified address, for example to adjust the settings of a speaker, the exception handling program is triggered to perform a specified task, such as to alert the program that the other application program accessed the device it intended to monitor.

21 Claims, 6 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 6

☐ 15. Document ID: US 5627992 A

L18: Entry 15 of 26 File: USPT May 6, 1997

US-PAT-NO: 5627992
DOCUMENT-IDENTIFIER: US 5627992 A
** See image for Certificate of Correction **

TITLE: Organization of an integrated cache unit for flexible usage in supporting microprocessor operations

DATE-ISSUED: May 6, 1997

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Baror; Gigy	Austin	TX		

US-CL-CURRENT: 711/133; 711/134, 711/142, 711/143, 711/144, 711/145, 711/205

ABSTRACT:

A computer system having a cache memory subsystem which allows flexible setting of caching policies on a page basis and a line basis. A cache block status field is provided for each cache block to indicate the cache block's state, such as shared or exclusive. The cache block status field controls whether the cache control unit operates in a write-through write mode or in a copy-back write mode when a write hit access to the block occurs. The cache block status field may be updated by either a TLB write policy field contained within a translation look-aside buffer entry which corresponds to the page of the access, or by a second input independent of the TLB entry which may be provided from the system on a line basis.

32 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

☐ 16. Document ID: US 5596764 A

L18: Entry 16 of 26 File: USPT Jan 21, 1997

US-PAT-NO: 5596764
DOCUMENT-IDENTIFIER: US 5596764 A

TITLE: Debug mechanism for parallel-operating DSP module and CPU core

DATE-ISSUED: January 21, 1997

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Intrater; Gideon	Tel-Aviv			IL
Katzri; Lior	Ramat-Aviv			IL
Viner; Omri	Hod Hasharon			IL
Levitan; Raya	Givataim			IL
Tzadik; Yehezkel	Hedera			IL

US-CL-CURRENT: 712/34; 712/227, 714/34, 714/38

ABSTRACT:

An integrated data processing system includes a shared internal bus for transferring both instructions and data. A shared bus interface unit is connected to the shared internal bus and connectable via a shared external bus to a shared external memory array such that

instructions and data held in the shared external memory array are transferrable to the shared internal bus via the shared bus interface unit. A general purpose (GP) central processing unit (CPU) is connected to the shared internal bus for retrieving GP instructions. The GP CPU includes an execution unit for executing GP instructions to process data retrieved by the GP CPU from the shared internal bus. A digital signal processor (DSP) module connected to the shared internal bus, the DSP module includes a signal processor for processing an externally-provided digital signal received by the DSP module by executing DSP command-list instructions. Execution of DSP command-list code instructions by the DSP module is independent of and in parallel with execution of GP instructions by the GP CPU. A shared internal memory that holds command-list code instructions and is connected for access by the DSP module for retrieval of command-list code instructions for execution by the DSP module and for access by the GP CPU for storage and retrieval of instructions and data.

5 Claims, 51 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 40

☐ 17. Document ID: US 5530804 A

L18: Entry 17 of 26 File: USPT Jun 25, 1996

US-PAT-NO: 5530804
DOCUMENT-IDENTIFIER: US 5530804 A

TITLE: Superscalar processor with plural pipelined execution units each unit selectively having both normal and debug modes

DATE-ISSUED: June 25, 1996

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Edgington; Gregory C.	Scottsdale	AZ		
Circello; Joseph C.	Phoenix	AZ		
McCarthy; Daniel M.	Phoenix	AZ		
Duerden; Richard	Scottsdale	AZ		

US-CL-CURRENT: 714/30; 703/28, 712/23, 712/43

ABSTRACT:

A processor (10) has two modes of operation. One mode of operation is a normal mode of operation wherein the processor (10) accesses user address space or supervisor address space to perform a predetermined function. The other mode of operation is referred to as a debug, test, or emulator mode of operation and is entered via an exception/interrupt. The debug mode is an alternate operational

mode of the processor (10) which has a unique debug address space which executes instructions from the normal instruction set of the processor (10). Furthermore, the debug mode of operation does not adversely affect the state of the normal mode of operation while executing debug, test, and emulation commands at normal processor speed. The debug mode is totally non-destructive and non-obtrusive to the "suspended" normal mode of operation. While in debug mode, the existing processor pipelines, bus interface, etc. are utilized.

42 Claims, 12 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 10

☐ 18. Document ID: US 5228131 A

L18: Entry 18 of 26 File: USPT Jul 13, 1993

US-PAT-NO: 5228131
DOCUMENT-IDENTIFIER: US 5228131 A

TITLE: Data processor with selectively enabled and disabled branch prediction operation

DATE-ISSUED: July 13, 1993

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Ueda; Tatsuya	Itami			JP
Yoshida; Toyohiko	Itami			JP

US-CL-CURRENT: 712/240

ABSTRACT:

The data processor related to the invention enables to designate whether the branch prediction mechanism itself should be activated or not for a conditional branch instruction, and the data processor enables to initialize branch history as required and also designates activation or inactivation of the branch prediction mechanism by setting a specific value to a specific bit of an exclusive usable register by software means. Also when a specific instruction is executed, the data processor automatically clears the branch history. As a result, in the event when the data processing efficiency is adversely declined by application of branch prediction mechanism or when monitoring external address bus, the branching prediction mechanism can be inactivated by setting the predetermined value to the exclusive usable register. Likewise, when the reliability of the branch history lowers due to such as variation in the program running condition, the data processor is capable of clearing the branch history by writing a specific value into the exclusive usable register, and when

executing a specific instruction which varies the program executing condition, branch history is automatically cleared.

7 Claims, 9 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 8

☐ 19. Document ID: US 5185878 A

L18: Entry 19 of 26 File: USPT Feb 9, 1993

US-PAT-NO: 5185878
DOCUMENT-IDENTIFIER: US 5185878 A

TITLE: Programmable cache memory as well as system incorporating same and method of operating programmable cache memory

DATE-ISSUED: February 9, 1993

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Baror; Gigy	Austin	TX		
Johnson; William M.	San Jose	CA		

US-CL-CURRENT: 711/123; 711/130, 711/146, 711/169

ABSTRACT:

Methods and apparatus are disclosed for realizing an integrated cache unit (ICU) comprising both a cache memory and a cache controller on a single chip. The novel ICU is capable of being programmed, supports high speed data and instruction processing applications in both Reduced Instruction Set Computers (RISC) and non-RISC architecture environments, and supports high speed processing applications in both single and multiprocessor systems. The preferred ICU has two buses, one for the processor interface and the other for a memory interface. The ICU support single, burst and pipelined processor accesses and is capable of operating at frequencies in excess of 25 megahertz, achieving processor access times of two cycles for the first access in a sequence, and one cycle for burst mode or piplined accesses. It can be used as either an instruction or data cache with flexible internal cache organization. A RISC processor and two ICUs (for instruction and data cache) implements a very high performance processor with 16k bytes of cache. Larger caches can be designed by using additional ICUs which, according to the preferred embodiment of the invention, are modular. Further features include flexible and extensive multiprocessor support hardware, low power requirements, and support of a combination of bus watching, ownership schemes, software control and hardware control schemes which may be used with the novel ICU to achieve cache consistency.

37 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 20. Document ID: US 5136691 A

L18: Entry 20 of 26 File: USPT Aug 4, 1992

US-PAT-NO: 5136691
DOCUMENT-IDENTIFIER: US 5136691 A
** See image for Certificate of Correction **

TITLE: Methods and apparatus for caching interlock variables in an integrated cache memory

DATE-ISSUED: August 4, 1992

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Baror; Gigy	Austin	TX		

US-CL-CURRENT: 711/139; 711/128, 711/142

ABSTRACT:

Methods and apparatus are disclosed for supporting the caching of interlock variables in cache memory units employed in multiprocessor and/or multitasking environments. The preferred embodiment of the invention includes methods and apparatus for selectively treating interlock variables as cachable or non-cachable. The disclosed methods and apparatus are suitable for supporting high speed data and instruction processing applications in both RISC and non-RISC architecture environments, can be integrated on a single chip and allows for better performance and utilization of the computer system bus structure since most of the interlock variable accesses are faster and do not appear on the memory bus (only in the cache).

45 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

Generate Collection | Print

Terms	Documents
L11	26

Display Format: REV Change Format

Previous Page Next Page

Search Results

Search Results for: **[debug circuit]**
Found **2** of **120,398 searched.**

Search within Results

Sort by: Title Publication Publication Date Score Binder

Results 1 - 2 of 2 short listing

1 Advances in functional abstraction from structure 77%
Richard H. Lathrop , Robert J. Hall , Gavan Duffy , K. Mark Alexander , Robert S. Kirk
Proceedings of the 25th ACM/IEEE conference on Design automation June 1988
FUNSTRUX has been extended to extract behavioral level models for a commercial simulator directly from a circuit netlist. Recent advances include: a retargetable code generation mechanism; an object-oriented control structure; handling of initialization values; and improved run-time and space requirements of the abstraction process. We discuss some of the issues that arise in translating from LISP to 'C' and from one functional paradigm to another.

2 An overview of Motorola's PowerPC simulator family 77%
William Anderson
Communications of the ACM June 1994
Volume 37 Issue 6

Results 1 - 2 of 2 short listing

Search Results

Search Results for: **[debug module]**
Found **8** of **120,398 searched.**

Search within Results

> Advanced Search

> Search Help/Tips

Sort by: Title Publication Publication Date Score Binder

Results 1 - 8 of 8 short listing

1 FSDS-Fairchild Software Development System 82%
Harley Mathews , Kam Li , John Katsaros
Proceedings of the 1975 annual conference January 1975
The Fairchild F8 Software Development System (FSDS) runs on the Fairchild F8 Microprocessor. Designed to ease the burden of developing F8 based microprocessor systems, FSDS enables testing programs in a real world environment. The FSDS system provides for generating, editing and maintaining source files, assembling user's programs, and executing routines using F8 hardware. This system was developed both for in-house Fairchild applications as well as customer based development programs.

2 A VHDL SGRGAM model for the validation environment of a high performance graphic processor 80%
M. G. Wahl , H. Völkel
Proceedings of the conference on Design, automation and test in Europe February 1998
To validate the functionality of a new highly complex graphics processor described in VHDL the working environment of the processors has to be modelled. In some cases appropriate models for the external components are commercially available, in other cases these models have to be created. In this paper a general memory model for SGRAMs is presented which had to be implemented to have a flexible simulation environment for a high speed graphics processor at hand. Key features are the generality, t ...

3 Architecture of a massively parallel processor 77%
Kenneth E. Batcher
Proceedings of the 7th annual symposium on Computer Architecture May 1980
The massively parallel processor (MPP) system is designed to process satellite imagery at high rates. A large number (16,384) of processing elements (PE's) are configured in a square array. For optimum performance on operands of arbitrary length, processing is performed in a bit-serial manner. On 8-bit integer data, addition can occur at 6553 million operations per

second (MOPS) and multiplication at 1861 MOPS. On 32-bit floating-point data, addition can occur at 430 MOPS and multiplication ...

4 Estimation of speed, area, and power of parameterizable, soft IP 77%
Jagesh Sanghavi , Albert Wang
Proceedings of the 38th conference on Design automation June 2001
We present a new approach to estimate speed, area, and power of a parameterizable, soft IP. By running the ASIC implementation flow only on selected configurations, we predict the performance for any arbitrary configuration. We exploit performance function decomposability to address the combinatorial explosion challenge. The estimator has been used successfully to configure Xtensa processor cores for numerous embedded SOC designs.

5 Display development system: a successful Ada application 77%
Robin R. Miller , Mary Ann Dodge
Proceedings of the third annual Washington Ada symposium on Ada: Ada use in focus : practical lessons in perspective March 1986

6 Linux kernel Internals 77%
Linux Journal January 1996

7 Architecture of a massively parallel processor 77%
Kenneth E. Batcher
25 years of the international symposia on Computer architecture (selected papers) August 1998

8 OCM—a monitoring system for interoperable tools 77%
Roland Wismüller , Jög Trinitis , Thomas Ludwig
Proceedings of the SIGMETRICS symposium on Parallel and distributed tools August 1998

Results 1 - 8 of 8 short listing

Your search matched **1** of **961219** documents.

A maximum of **1** results are displayed, **15** to a page, sorted by **Relevance** in **descending** order.
You may refine your search by editing the current search expression or entering a new one the text box.

Then click **Search Again**.

('debug information')and (circuit)

Search Again

Results:
Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**

1 **A module diagnosis and design-for-debug methodology based on hierarchical test paths**
Makris, Y.; Orailoglu, A.;
Defect and Fault Tolerance in VLSI Systems, 1999. DFT '99. International Symposium on , 1-3 Nov. 1999
Page(s): 339 -347

[Abstract] [PDF Full-Text (116 KB)] **IEEE CNF**

Your search matched **29** of **961219** documents.

A maximum of **29** results are displayed, **15** to a page, sorted by **Relevance** in **descending** order.
You may refine your search by editing the current search expression or entering a new one the text box.

Then click **Search Again**.

((debug)and (circuit)) and(processor)

Search Again

Results:
Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**

1 **Position statement: TAPs all over my chips**
Oakland, S.F.;
Test Conference, 2002. Proceedings. International , 7-10 Oct. 2002
Page(s): 1192

[Abstract] [PDF Full-Text (194 KB)] **IEEE CNF**

2 **Dual processor VHSIC MIL-STD-1750 A computer module**
Coulon, K.;
Aerospace and Electronics Conference, 1988. NAECON 1988.,
Proceedings of the IEEE 1988 National , 23-27 May 1988
Page(s): 54 -59 vol.1

[Abstract] [PDF Full-Text (600 KB)] **IEEE CNF**

3 **A case-study in the use of scan in microSPARC testing and debug**
Katz, J.;
Test Conference, 1994. Proceedings., International , 2-6 Oct. 1994
Page(s): 456 -460

[Abstract] [PDF Full-Text (368 KB)] **IEEE CNF**

4 **Considerations for implementing IEEE 1149.1 on system-on-a-chip integrated circuits**
Oakland, S.F.;
Test Conference, 2000. Proceedings. International , 3-5 Oct. 2000
Page(s): 628 -637

[Abstract] [PDF Full-Text (716 KB)] **IEEE CNF**

5 **On identifying indistinguishable path delay faults and improving diagnosis**
Tekumalla, R.C.; Davidson, S.;
Test Conference, 2002. Proceedings. International , 7-10 Oct. 2002
Page(s): 993 -1002

[Abstract] [PDF Full-Text (694 KB)] **IEEE CNF**

6 **Silicon symptoms to solutions: applying design for debug techniques**
Pyron, C.; Bangalore, R.; Belete, D.; Goertz, J.; Razdan, A.; Younger, D.;
Test Conference, 2002. Proceedings. International , 7-10 Oct. 2002
Page(s): 664 -672

[Abstract] [PDF Full-Text (636 KB)] **IEEE CNF**

7 **A user's view of MCM-D/C packaging: is it worth the trouble?**

Bartley, J.;
Electronic Components and Technology Conference, 1996. Proceedings., 46th , 28-31 May 1996
Page(s): 144 -148

[Abstract] [PDF Full-Text (848 KB)] **IEEE CNF**

8 **Knowledge-based electrical monitor approach using very large array yield structures to delineate defects during process development and production yield improvement**
Hammond, J.; Sery, G.;
Defect and Fault Tolerance on VLSI Systems, 1991. Proceedings., 1991 International Workshop on , 18-20 Nov. 1991
Page(s): 67 -80

[Abstract] [PDF Full-Text (568 KB)] **IEEE CNF**

9 **An in-circuit signal analyzer for mixed signal digital signal processor**
Beling, S.; Leary, K.; Yukna, G.;
Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on , 14-17 April 1991

Page(s): 1109 -1112 vol.2

[Abstract] [PDF Full-Text (504 KB)] **IEEE CNF**

10 **Embedded 6 bit flash converter design for digital stereo sound decoder**
Pathak, V.; Ritchie, K.; Kitchin, M.;
ASIC Conference and Exhibit, 1992., Proceedings of Fifth Annual IEEE International , 21-25 Sept. 1992
Page(s): 391 -395

[Abstract] [PDF Full-Text (484 KB)] **IEEE CNF**

11 **A chip to embedded system test process using IEEE 1149.1 boundary scan**
Kadaras, J.E.;
Electro/94 International. Conference Proceedings. Combined Volumes. , 10-12 May 1994
Page(s): 728 -732

[Abstract] [PDF Full-Text (232 KB)] **IEEE CNF**

12 **An integrated software platform for the design and DSP-based implementation of digital filters**
Filho, M.S.; Schneebeli, H.A.; Machado, A.C.;
Circuits and Systems, 1996., IEEE 39th Midwest symposium on , Volume: 2 , 18-21 Aug. 1996
Page(s): 868 -871 vol.2

[Abstract] [PDF Full-Text (516 KB)] **IEEE CNF**

13 **Pentium(R) Pro processor design for test and debug**
Carbine, A.; Feltham, D.;
Test Conference, 1997. Proceedings., International , 1-6 Nov. 1997
Page(s): 294 -303

[Abstract] [PDF Full-Text (932 KB)] **IEEE CNF**

14 **Testing the 400 MHz IBM generation-4 CMOS chip**
Foote, T.G.; Hoffman, D.E.; Huott, W.V.; Koprowski, T.J.; Robbins, B.J.; Kusko, M.P.;
Test Conference, 1997. Proceedings., International , 1-6 Nov. 1997
Page(s): 106 -114

[Abstract] [PDF Full-Text (796 KB)] **IEEE CNF**

15 **Testing embedded-core based system chips**
Zorian, Y.; Marinissen, E.J.; Dey, S.;
Test Conference, 1998. Proceedings. International , 18-23 Oct. 1998
Page(s): 130 -143

[Abstract] [PDF Full-Text (1340 KB)] **IEEE CNF**

1 2 [Next]

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search
Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting
No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

Your search matched **29** of **961219** documents.

A maximum of **29** results are displayed, **15** to a page, sorted by **Relevance** in **descending** order.
You may refine your search by editing the current search expression or entering a new one the text box.
Then click **Search Again**.

Search Again

Results:
Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**

16 **Design and implementation of the “G2” PowerPC™ 603e-embedded microprocessor core**
Hunter, C.; Gaither, J.;
Test Conference, 1998. Proceedings. International , 18-23 Oct. 1998
Page(s): 473 -479

[Abstract] [PDF Full-Text (700 KB)] **IEEE CNF**

17 **Proceedings Design, Automation and Test in Europe**
Design, Automation and Test in Europe, 1998., Proceedings , 23-26 Feb. 1998

[Abstract] [PDF Full-Text (780 KB)] **IEEE CNF**

18 **Design for testability: it is time to deliver it for Time-to-Market**
Dervisolu, B.;
Test Conference, 1999. Proceedings. International , 28-30 Sept. 1999
Page(s): 1102 -1111

[Abstract] [PDF Full-Text (1004 KB)] **IEEE CNF**

19 **The test and debug features of the AMD-K7™ microprocessor**
Wood, T.J.;
Test Conference, 1999. Proceedings. International , 28-30 Sept. 1999
Page(s): 130 -136

[Abstract] [PDF Full-Text (564 KB)] **IEEE CNF**

20 **Debug facilities in the TriMedia CPU64 architecture**
Vranken, H.;
Test Workshop 1999. Proceedings. European , 25-28 May 1999
Page(s): 76 -81

[Abstract] [PDF Full-Text (96 KB)] **IEEE CNF**

21 **A 780 MHz PowerPC™ microprocessor with integrated L2 cache**
Bearden, D.R.; Caffo, D.G.; Anderson, P.; Rossbach, P.; Iyengar, N.; Petersen, T.A.; Jen-Tien Yen;
Solid-State Circuits Conference, 2000. Digest of Technical Papers. ISSCC. 2000 IEEE International , 7-9 Feb. 2000
Page(s): 90 -91

[Abstract] [PDF Full-Text (222 KB)] **IEEE CNF**

22 **Test and on-line debug capabilities of IEEE Std 1149.1 in UltraSPARC™-III microprocessor**
Golshan, F.;
Test Conference, 2000. Proceedings. International , 3-5 Oct. 2000
Page(s): 141 -150

[Abstract] [PDF Full-Text (616 KB)] **IEEE CNF**

23 **Test and debug strategy of the PNX8525 Nexperia™ digital video platform system chip**
Vermeulen, B.; Oostdijk, S.; Bouwman, F.;
Test Conference, 2001. Proceedings. International , 30 Oct.-1 Nov. 2001
Page(s): 121 -130

[Abstract] [PDF Full-Text (877 KB)] **IEEE CNF**

24 **The manic depression of microprocessor debug**
Josephson, D.D.;
Test Conference, 2002. Proceedings. International , 7-10 Oct. 2002
Page(s): 657 -663

[Abstract] [PDF Full-Text (522 KB)] **IEEE CNF**

25 **Reusable embedded debugger for 32 bit RISC processor using the JTAG boundary scan architecture**
Dae-Young Jung; Sung-Ho Kwak; Moon-Key Lee;
ASIC, 2002. Proceedings. 2002 IEEE Asia-Pacific Conference on , 6-8 Aug. 2002
Page(s): 209 -212

[Abstract] [PDF Full-Text (293 KB)] **IEEE CNF**

26 **Application of scan hardware and software for debug and diagnostics in a workstation environment**
Dervisoglu, B.I.;
Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on , Volume: 9 Issue: 6 , June 1990
Page(s): 612 -620

[Abstract] [PDF Full-Text (800 KB)] **IEEE JNL**

27 **Rapid prototyping for DSP systems with multiprocessors**
Engels, M.; Lauwereins, R.; Peperstraete, J.A.;
Design & Test of Computers, IEEE , Volume: 8 Issue: 2 , June 1991
Page(s): 52 -62

[Abstract] [PDF Full-Text (944 KB)] **IEEE JNL**

28 **AVPGEN-A test generator for architecture verification**
Chandra, A.; Iyengar, V.; Jameson, D.; Jawalekar, R.; Nair, I.; Rosen, B.; Mullen, M.; Yoon, J.; Armoni, R.; Geist, D.; Wolfsthal, Y.;
Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , Volume: 3 Issue: 2 , June 1995
Page(s): 188 -200

[Abstract] [PDF Full-Text (1204 KB)] **IEEE JNL**

29 **A multigigahertz clocking scheme for the Pentium(R) 4 microprocessor**
Kurd, N.A.; Barkarullah, J.S.; Dizon, R.O.; Fletcher, T.D.; Madland, P.D.;
Solid-State Circuits, IEEE Journal of , Volume: 36 Issue: 11 , Nov. 2001
Page(s): 1647 -1653

[Abstract] [PDF Full-Text (371 KB)] **IEEE JNL**

[Prev] 1 2

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search
Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting
No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

CiteSeer Find: debug circuit Documents Citations

Searching for **PHRASE debug circuit**.
Restrict to: Header Title Order by: Citations Hubs Usage Date Try: Amazon B&N Google (RI) Google (Web) CSB DBLP
2 documents found. **Order: citations weighted by year.**

Mechanisms for Dynamically Changing Initiative in.. - Department (1996) (Correct) (1 citation)
Computer Initiative: Computer goal(fix_circuit)**debug**(led,off)goal(fix_circuit)**debug**(led,off)
debug(led,off)goal(fix_circuit)**debug**(led,off)goal(fix_circuit)**debug**(led,off)
In the implemented voice dialogue system "The **Circuit** Fix-it Shop" 19]the following dialogue
ftp.cs.duke.edu/pub/cig/papers/hics.ps.Z

Human–Computer Collaborative - Curry Guinn Department (Correct)
Stack Initiative: Computer Initiative: Computer **debug**(led,off)observe(switch)Initiative: Computer
Problem-Solving Stack Initiative: Computer **debug**(led,off)Initiative: Computer
In the implemented voice dialogue system "The **Circuit** Fix-it Shop" Smith et al.1992 Smith and
www.cs.duke.edu/~cig/papers/ACL96.PS

Try your query at: Amazon Barnes & Noble Google (RI) Google (Web) CSB DBLP